

Do Sensor Networks need Mobile MAC Protocols?

Navid Hassanzadeh¹, Olaf Landsiedel^{2,4}, Frederik Hermans³, Olof Rensfelt³
and Thiemo Voigt¹

¹ Swedish Institute of Computer Science (SICS), Kista, Sweden

² Chalmers University of Technology, Sweden

³ Uppsala University, Sweden

⁴ KTH Royal Institute of Technology, Sweden

Abstract. There exists a number of MAC protocols targeted for mobile scenarios. These include MMAC, MS-MAC and AM-MAC. These MAC protocols have in common that they seem to be evaluated only in simulation. This might indicate that these MAC are either too complex to use or they are not needed, at least for data collection, the major task of sensor networks. In this paper we show that extending a traditional data collection protocol with lightweight, carefully selected mechanisms is sufficient to provide reliable data collection at low energy cost for mobile sensor networks where both sinks and sources move.

1 Introduction

There are quite a few MAC layers specifically designed for mobile scenarios. These include MMAC [1], MS-MAC [2] and AM-MAC [3]. All of these MAC protocols are evaluated by simulation and it is unclear if implementations for real sensor node hardware exist. Given the complexity of some of these protocols and the absence of implemenations that work on real sensor nodes, we wonder if these mobile MAC protocols are really needed. We investigate this question for mobile scenarios where mobile sink nodes collect data from mobile data sources. We target scenarios where data is collected in real-time. This is in contrast to approaches that use mechanisms from delay-tolerant networking such as ZebraNet [4] or RatPack [5] where data is collected for off-line analysis and hence longer delays do not matter.

In this paper we advocate a different approach to tackle the challenging issue of mobile data collection. Rather than designing a new MAC protocol, we enhance an existing data collection protocol for static networks with lightweight mechanisms to improve performance in mobile scenarios. Towards this end, we modify the Contiki Collect protocol [6], a protocol similar to the Collection Tree Protocol (CTP) [7] for Tiny OS, to make it more suitable for mobile scenarios. We enhance Contiki Collect with mechanisms to detect and repair loops since these occur more often in mobile than in static scenarios. Furthermore, we enable nodes to quickly find new parents as nodes often move out of range in mobile scenarios. We provide an implementation for the Contiki operating system that we call Mobile Collect.

We evaluate Mobile Collect by both simulation and in a testbed that includes mobile robots. We also perform simulations with a random waypoint mobility model. Our experiments show that Mobile Collect performs very well in such a scenario. Even in scenarios with very high mobility (2 to 8 m/s), Mobile Collect still achieves a delivery rate of 70% and more at a low energy consumption of 10 mJ per received packet. Note that, for example, the simulations in MMAC were performed with an average speed of only 0.1 m/s [1]. Given these results, we conclude that in many scenarios with mobile nodes, there is no eminent need for new MAC protocols specifically designed for mobility.

2 Design and Implementation

As mentioned above we enhance Contiki Collect with two types of mechanisms. First, we enable nodes to more quickly find new parents as in mobile scenarios nodes often move out of range. Second, we need mechanisms to detect and repair loops since these occur more often in mobile than in static scenarios. Additionally, we integrate routing beacons into the receiver initiated MAC-layer.

MAC Layer Beacons: Topology dynamics triggered by node mobility require nodes to frequently announce their presence and routing metrics to neighboring nodes. To avoid additional cost from this beaconing, Mobile Collect integrates its beacons into the MAC layer: In each probe of its receiver initiated MAC [8] a node also announces its current routing metric. As these probes are transmitted on each wakeup of our duty-cycled nodes, Mobile Collect achieves a high rate of routing beacons with essentially no additional cost.

Parent Switch on Timeout: In mobile scenarios, we argue that a routing timeout indicates that the target node has disappeared from the communication range of the sending node. Thus, instead of punishing the timed-out route by slightly increasing its routing metric ETX, as done commonly in routing protocols, we increase the ETX to the maximum value which enforces a parent switch. Hence, the source node disconnects from its parent and (1) connects to alternate parents in its neighbor table, or (2) if no alternative parents are available it uses beaconing as fall-back to discover new parents.

Avoiding Routing Loops: The dynamic topology caused by the mobility of nodes and the agile parent change in Mobile Collect increase the risk of routing loops. Mobile Collect extends Contiki Collect by enabling a node to track the parents of all its neighbors. Thus, nodes in Mobile Collect announce the IDs of their parents in their routing beacons. This allows us to implement classic loop suppression mechanisms such as triangle suppression [9].

Our evaluation shows that we achieve high reliability and energy efficiency for data collection in mobile settings with these lightweight mechanisms.

3 Evaluation

We perform simulations in COOJA [10] and use BonnMotion [11] to generate the mobility scenarios. We simulate 50 nodes out of which three are sink nodes.

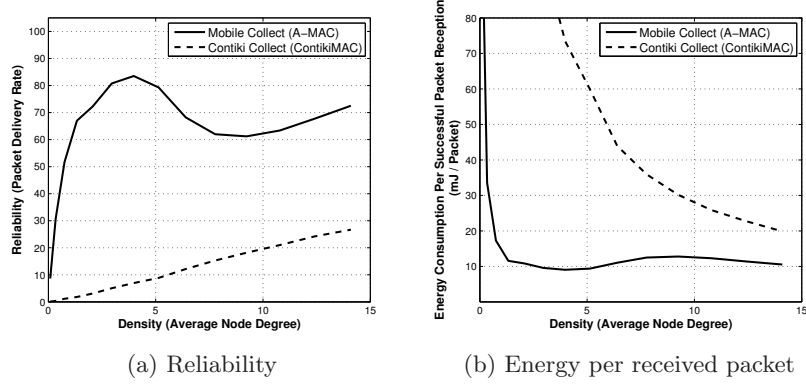


Fig. 1: Reliability and energy performance of Mobile and Contiki Collect under the random waypoint mobility model. Mobile Collect performs very well.

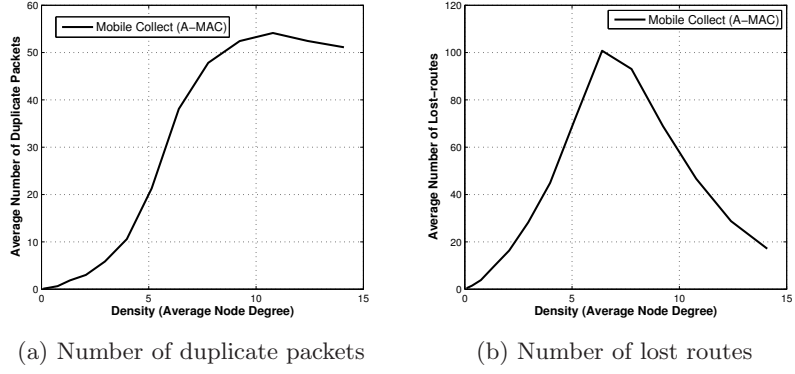


Fig. 2: Microbenchmarks that explain the performance of Mobile Collect

We use a random waypoint mobility model with a node speed between 2 and 8 m/s.

Our results are shown in Figure 1. Mobile Collect achieves much higher packet delivery rates at lower energy consumption than Contiki Collect. Figure 1a shows that the packet delivery rate (PDR) first increases, then for node degrees between 4 and 8 decreases before it increases again. For very low densities, the network is sparse, so nodes do not find forwarders to forward their packets to the sink. This improves with higher density, but as shown in Figure 2a the number of duplicate packets then increases which is caused by packets not being acknowledged and larger than triangle loops. The negative trend does not continue for medium node degrees as shown in Figure 2b: the number of lost routes decreases as nodes can more easily find forwarders as the network becomes more dense. Figure 1b shows much better energy consumption per received byte for Mobile Collect than for Contiki Collect mainly because the PDR is much higher and there is less energy

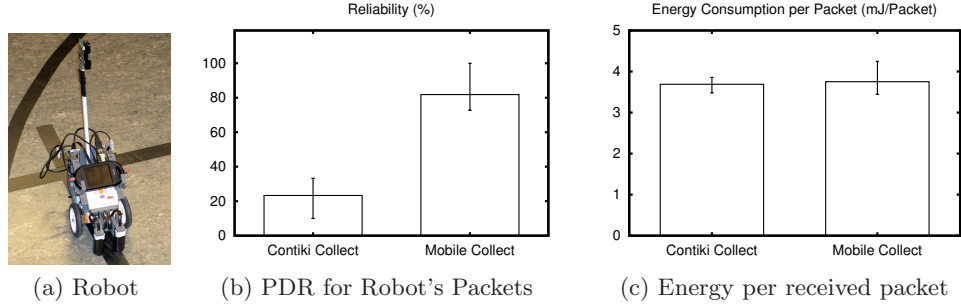


Fig. 3: Mobile Collect delivers more packets from the mobile robot at the same energy cost as Contiki Collect.

wasted on packets that do not make it to the sink. This is very apparent for Contiki Collect when the network is sparse.

We depict results from a robot sensor network in Figure 3. In this scenario, one robot acting as data source moves from one cluster of static nodes to a second cluster of static nodes. Both clusters have one sink each. Mobile Collect shows much better performance than Contiki Collect since it *(i)* is able to buffer data when moving from the first cluster to the second and *(ii)* with Mobile Collect it is able to find new routes quickly as it comes in range of the second cluster. The packet loss is mainly due to the early determination of the measurements: with Mobile Collect all packets have eventually arrived at the sink.

4 Conclusions

Given the promising results of Mobile Collect in demanding mobility scenarios and on real hardware, we doubt that sensor networks really need MAC layers specifically designed for mobile scenarios.

5 Acknowledgements

This work has been partly funded by SSF, VR, the SICS Center for Networked Systems (CNS), the Uppsala VINN Excellence Center for Wireless Sensor Networks WISENET, and partly supported the FP7 NoE CONET. CNS is funded by VINNOVA, SSF, KKS, ABB, Ericsson, Saab SDS, TeliaSonera, T2Data, Vendolocus and Peerialism. WISENET is funded by VINNOVA, Uppsala University, Banverket, CRL Sweden, FOI, Imego, JonDeTech, Pricer, SenseAir, SICS, TNT-Elektronik, TermoSense, VTT, Upwis, ÅAC Microtec, and WiseNet Holding. The authors thank Gunnar Karlsson (KTH) and Peter Stenlund (Vendolocus) for their input.

References

1. M. Ali, T. Suleman, and Z.A. Uzmi. MMAC: A mobility-adaptive, collision-free mac protocol for wireless sensor networks. In *IPCC*, Phoenix, USA, April 2005.
2. D. Zhang, Q. Li, X. Zhang, and X. Wang. De-ass: An adaptive mac algorithm based on mobility evaluation for wireless sensor networks. In *WiCOM*, 2010.
3. S.C. et al. Choi. An adaptive mobility-supporting mac protocol for mobile sensor networks. In *Vehicular Technology Conference*, May 2008.
4. T. Liu et al. Implementing software on resource-constrained mobile sensors: Experiences with Impala and ZebraNet. In *ACM MobiSys*, June 2004.
5. J.A.B. Link, K. Wehrle, O. Osechas, and J. Thiele. Ratpack: Communication in a sparse dynamic network. In *ACM Sigcomm*, Seattle, USA, August 2008.
6. J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, M. Durvy, JP Vasseur, A. Terzis, A. Dunkels, and D. Culler. Beyond Interoperability: Pushing the Performance of Sensornet IP Stacks. In *ACM SenSys*, Seattle, USA, November 2011.
7. O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *ACM SenSys*, Berkeley, USA, November 2009.
8. P. et al. Dutta. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *ACM SenSys*, November 2010.
9. C. L. Hedrick. RFC 1058: Routing information protocol, June 1988.
10. F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with cooja. In *SenseApp*, November 2006.
11. N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn. Bonnmotion: a mobility scenario generation and analysis tool. In *SIMUTools*, March 2010.